A Note on the Empirical Evaluation of Security Margins against Algebraic Attacks
(with Application to Low Cost-Ciphers LED and Piccolo)
- Extended Asbtract -

Vincent Grosso[1]*, Christina Boura[2,3], Benoît Gérard[1]**, François-Xavier Standaert[1]***

[1] UCL Crypto Group, place du Levant 2, 1348 Louvain-la-Neuve, Belgium.
[2] Equipe SECRET, INRIA Rocquencourt, B.P. 105, F-78153 Le Chesnay Cedex, France.
[3] Gemalto - 6, rue de la Verrerie - 92447 Meudon sur Seine, France.

Abstract. Algebraic attacks are an important class of cryptanalytic techniques. Yet, precisely estimating the security margins that a block cipher may provide against them is generally difficult, as sound theoretical tools are missing for this purpose. Therefore, most recent block cipher proposals combine different heuristic arguments in order to argue about their practical security against such attacks. In this paper, we discuss the relevance and correlation of these arguments, with a practical case-study based on the lightweight ciphers LED and Piccolo.

1 Introduction

The design of modern block ciphers usually comes with arguments of security against several cryptanalysis techniques. These arguments typically include the evaluation of statistical attacks such as linear and differential cryptanalysis [2, 18], and the investigation of structural properties leading to integral or slides attack [4, 17]. A number of well understood heuristic tools are available for this purpose. For example, the wide-trail strategy can be used to design block ciphers that are practically secure against statistical attacks [12]. One interesting feature of these heuristic tools is that they do not only provide security, but also allow the evaluation of (informal) bounds against certain categories of attacks. As a result, they can be used to compare different algorithms.

For other types of attacks though, and in particular for the algebraic cryptanalysis that we consider in this paper, security analyzes are hardly as systematic. This may sound counterintuitive, as algebraic complexity is known to be a central criteria for block cipher security since the seminal work of Shannon [23]. Yet, the discussion of algebraic attacks usually comes late (if at all) in block cipher specifications.

One possible reason of this situation is that the complexity of algebraic attacks is hard to evaluate [7, 8, 11, 19]. It is also not simple to interpret successful attacks against weakened versions of a block cipher. Yet, it remains that instances of (admittedly weak) block ciphers that are actually broken with algebraic attacks exist in the literature (e.g. the Keeloq and MiFare ciphers [9, 10]). Besides, it has also been shown that the algebraic complexity of a block cipher has a significant impact in the context of algebraic side-channel attacks [21, 22]. Hence, in view of the recent design of numerous lightweight block ciphers for constrained applications [14], it becomes increasingly interesting to understand the extent to which these new proposals with simplified structure and low implementation cost remain sufficiently robust against algebraic attacks.

* PhD student supported by the ERC project 280141 - CRASH.
** Postdoctoral researcher supported by Walloon region MIPSs project.
*** Associate researcher of the Belgian Fund for Scientific Research (FNRS-F.R.S.).

The security of a block cipher against such cryptanalyses can be argued with different types of metrics. For example, the complexities of the systems of equations representing different block ciphers have been compared in [3]. Another solution is to estimate the number of block cipher rounds needed to reach maximum algebraic degree [5]. More recently, cube testers have been proposed as another systematic alternative to construct algebraic distinguishers [1]. Eventually, the most direct approach is to investigate the complexity of solving a block cipher system of equations, e.g. with a SAT solver or Groebner basis tools [15]. However in this last case, directly solving the system of equations of a cipher should always be impossible (or would be the sign of a very weak design). This raises the question of which reduced versions of a cipher can be used to meaningfully argue about security margins against algebraic cryptanalysis.

In this paper, we consider the lack of comprehensive tools for the evaluation of algebraic attacks and discuss the relevance and limitations of a combined approach. Namely, we mix the estimation of informal criteria such as the system of equations size or the algebraic degree of a block cipher, with heuristic criteria such as the solving time of attacks against different versions of the target cipher. For this purpose, we consider attacks against full ciphers with variable guessing strategies, and attacks against reduced ciphers with fixed guessing strategy. Next, and taking the example of the block ciphers LED and Piccolo [16, 24], we discuss the extent to which these criteria lead to similar intuitions and can be used to estimate security margins against algebraic cryptanalysis. Doing so, we introduce a metric of "Equivalent Encryption Time" (EET) which essentially corresponds to the encryption speed that has to be reached for an exhaustive key search to be more efficient than an algebraic cryptanalysis. Let t be the median time needed for an algebraic attack to succeed, and n be the number of key bits to find, EET is defined as $t/2^n$. One interesting feature of this metric is that it allows comparing the security of different algorithms against algebraic attacks (i.e. their advantage over exhaustive search), independent of their encryption time. We use it to comment on the larger security margins of LED compared to Piccolo, and highlight observations regarding the impact of using a SAT solver in security evaluations.

Note. Because of space constraints, some details and parts of the background of this extended abstract have been deferred to a full version available online.

2 Background

The description of the block cipher LED and Piccolo is given in the full version, together with a brief description of an links towards references on algebraic cryptanalysis.

2.1 Algebraic degree of a Boolean function

From a mathematical point of view, a cipher E is a vectorial Boolean function of dimension n having m variables:

$$\begin{aligned}\mathsf{E} : (X, K) &\mapsto Y = \mathsf{E}(X, K),\\ \mathbb{F}_2^m &\to \mathbb{F}_2^n.\end{aligned}$$

That is, each of the n coordinates is a Boolean function with m variables that should not be distinguishable from a randomly generated Boolean function. In general, any non-random behavior of any combination of coordinates can be the sign of a weakness and may be exploited in an attack (e.g. linear cryptanalysis [18], etc.). In the case of algebraic attacks, the degree of its Boolean functions is a good indicator for the strength of a block cipher. It is defined as follows.

Definition 1. *Algebraic degree. Let g be a Boolean function from $\mathbb{F}_2^m$ into $\mathbb{F}_2$. Such a Boolean function can be represented using its algebraic normal form (ANF):*

$$g(x_1, \ldots, x_m) = \sum_{(u_1,\ldots,u_m)\in\mathbb{F}_2^m} a_{(u_1,\ldots,u_m)} \prod_{i=1}^{m} x_i^{u_i}.$$

Such representation is unique and allows a simple definition of the degree of g:

$$\deg(g) \triangleq \max_{(u_1,\ldots,u_m)\in\mathbb{F}_2^m} \left\{ u = \sum_{i=1}^{m} u_i, a_{(u_1,\ldots,u_m)} \neq 0 \right\}.$$

In the case of a vector Boolean function $G = (g_1, \ldots, g_n)$, the degree is defined as the maximum degree of its coordinates:

$$\deg(G) \triangleq \max_{1\leq i\leq n} \deg(g_i).$$

In general, the state size of recent ciphers makes the explicit computation of the ANF for any coordinate of the cipher an intractable problem (a similar comment holds for hash functions). This situation motivated the derivation of bounds to estimate the algebraic degree as part of the security evaluation of a cryptographic primitive.

Taking the example of LED and Piccolo, a natural question is to determine how the degree of these ciphers evolves with the number of round iterations[1]. The most obvious way to do this, is to use what is generally called the *trivial bound*. It consists in bounding the degree of a round permutation of degree d after r iterations by d^r. This trivial bound usually gives good results when the number of rounds is relatively small, but fails as this number increases. For this purpose, a better estimation is needed.

Let us now denote the round functions of LED and Piccolo as $F = L' \circ S \circ L$, with S the non-linear S-box layer and L a linear (over $\mathbb{F}_2^n$) diffusion layer (i.e. two consecutive S-box layers will be separated by the linear function $L \circ L'$). Due to the mixing effect of the linear layer, the following quantity will be of interest for "chaining" rounds.

Definition 2. *Let $G = (g_1, \ldots, g_n)$ be a vector Boolean function of dimension n. Then we denote by $\delta_k(G)$ the maximal degree of the product of at most k coordinates of G:*

$$\delta_k(G) \triangleq \max_{\substack{I\subset\{1,\ldots,n\}\\ \#I\leq k}} \deg\left(\prod_{i\in I} g_i\right).$$

We now present a former result from [6]: it aims to provides a better approximation for the algebraic degree of a block cipher than the trivial bound.

[1] Since for both ciphers, the key addition corresponds to a bitwise XOR with a constant, the algebraic degree of the cipher can be computed independent of the key scheduling part.

Theorem 1. *If a vector Boolean function S from $\mathbb{F}_2^n$ to $\mathbb{F}_2^n$ consists in the concatenation of smaller permutations from $\mathbb{F}_2^{n'}$ to $\mathbb{F}_2^{n'}$, then for any vector Boolean function G, the degree of $G \circ S$ is upper-bounded by:*

$$\deg(G \circ S) \leq n - \frac{n - \deg(G)}{\gamma(S)}, \text{ where } \gamma(S) \triangleq \max_{1 \leq i \leq n'-1} \frac{n' - i}{n' - \delta_i(S)}.$$

3 Estimating the algebraic degree of LED and Piccolo

In this section, we estimate the algebraic degree of LED and Piccolo. The goal of this estimation is to investigate possible links between the algebraic degree and the resistance of these block ciphers against the SAT-based algebraic attacks in Section 4.

3.1 On the algebraic degree of LED

The non-linear layer of LED consists in the parallel application of the PRESENT S-box. This S-box is a 4-bit permutation with algebraic degree 3. Thus, the degree of one round is also 3. We use Theorem 1 to derive bounds on the degrees for more rounds. Let us recall this bound for this particular instance where $\gamma(S) = 3$:

$$\deg(F_r) \leq 64 - \frac{64 - \deg(F_{r-1})}{3}.$$

It directly gives the results in Table 1. As expected, it is not tight for small degrees.

Table 1. Bounds on the algebraic degree of r-round LED.

Numbers of rounds	Trivial bound	Theorem 1
1	**3**	-
2	**9**	45
3	**27**	47
4	63	**51**
5	63	**59**
6	63	**62**
7	**63**	**63**

3.2 On the algebraic degree of Piccolo

The only source of non-linearity of Piccolo is the 16-bit-to-16-bit function F that is composed of two identical S-box layers separated by a matrix multiplication. As for LED, the algebraic degree of this function can be estimated and extended to the full cipher using Theorem 1. Yet, contrary to LED, this estimation for Piccolo is made more difficult because of non-uniformities in the degrees of the F function output bits. In the following, we only provide the final results of our investigations, in Table 2. It contains the trivial bound, a straightforward application of Theorem 1 and a refined application of Theorem 1, based on an analysis of the degree for 2 rounds of Piccolo. Let us notice that the first non-trivial bound is far from being tight according to the gap we can observe with entries in the last column. This suggests that the algebraic degree of Piccolo may be harder to estimate than the one of LED. It is likely that even the last column does not reflect the actual behavior of this algebraic degree. *Details about this degree investigation are given in the full version of the paper.*

Table 2. Bounds on the algebraic degree of r-round Piccolo.

Numbers of rounds	Trivial bound	Theorem 1	Theorem 1 + 2-round degree analysis
1	**9**	-	-
2	63	47	**29**
3	63	60	**52**
4	63	63	**62**
5	**63**	**63**	**63**

4 Experimenting SAT-based algebraic attacks

Our algebraic cryptanalysis experiments are based on a SAT solver. We used the MiniSat v1.14 SAT solver [13] that is an open-source tool rewarded in different SAT competitions. Exploiting it requires describing the target cipher as a CNF, i.e. a conjunction of disjunction of variables. For this purpose, the straightforward strategy would be to express every ciphertext bit directly in function of the plaintext variables. However, this implies the apparition of numerous high degree monomials that are hardly managed by the solver. To overcome this limitation, the usual approach is to introduce intermediate literals in the cipher description (details are given next).

Since recovering the full cipher keys without additional information than a plaintext/ciphertext pair is (hopefully) difficult, our experiments were performed giving some key bit values as extra information to the solver. Depending on the experiments, this number of bits provided may differ. We will refer as *number of unknown key bits* the remaining number of key bit variables in the system after providing the extra information. Note that we chose to fix the first key bits of the master keys.

4.1 On the size of the CNF representation

In this section we focus on the complexity of the representation of both ciphers. Since the representation may strongly influence the resolution time, we tried to build systems having similar structures. Hence, we used the same construction method for LED and Piccolo. For the 64-bit key full-version of LED, the CNF representation has been obtained by adding intermediate literals before and after both the key additions and the AddConstants operations, and after the SubCells operations. As a result, we obtained a system of approximately 70.000 equations in 12.000 variables with at most 12 literals per clause. For the 80-bit key full-version of Piccolo, the CNF representation has been obtained by adding intermediate literals before and after each S-box in the F-function, and after the bit-wise addition between the state and the round keys. Concerning the key scheduling, we added literals for all the sub-keys. As a result, we obtained a representation of the Piccolo cryptosystem with 6.000 variables used in 50.000 equations. There are at most 8 variables per clause. For both ciphers, the representation of the S-box has been obtained with the same method, and gives 64 equations of 5 literals each and has 8 literals. We can see that the LED representation requires more equations than Piccolo (respectively 70.000 and 50.000), while the number of variables in the LED representation is twice the number of variables in Piccolo. Moreover, we notice that the literals are more connected in the LED representation than in the Piccolo representation. This may suggest that LED is more robust than Piccolo against algebraic attacks, at least when deriving representations in such a straightforward fashion.

4.2 Attacking the full-version with variable key-guess sizes

We now consider the evolution of the resolution time of the system as a function of the number of the key bits unknown to the solver. The target ciphers are the full-version of the algorithms proposed at CHES. The representation used are the one described in Section 4.1 and, as mentioned earlier, we have fixed the first key bit values. We aimed at comparing resolution times for a number of unknown key bits ranging from 4 to 18, which translates in providing 46 to 60-bit values (resp. 62 to 76 bits) to the solver for recovering the full key of LED (resp. Piccolo). The experimental results obtained are provided in Figure 1, where the curves represent the evolution of the median solving time as a function of the number of key bits unknown to the solver. As expected, we can observe that the resolution time grows as an exponential function of the number of unknown bits for both ciphers. We also notice that the Piccolo curve is shifted by 2 bits on the x-axis compared to the LED curve. Hence, this metric again suggests that Piccolo could be slightly weaker than LED against algebraic cryptanalysis.

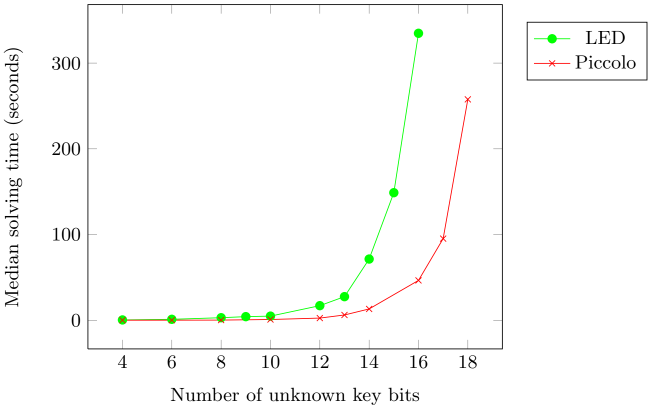

Fig. 1. Solving time vs number of unknown key bits for LED and Piccolo.

Next, we translated these median solving times into EET. Intuitively, the EET represents the encryption speed that should be reached by an implementation of the block ciphers, for the exhaustive search to be more efficient than the SAT solver based cryptanalysis. Results are plotted in Figure 2 and lead to the following additional observations. First, and in both plots, two different parts can be observed: the EET initially decreases up to a certain point, where it then becomes stable. While having the same shape, both curves differ in the point where the EET becomes stable. The EET for LED stops decreasing when more than 10 key bits are unknown (stabilizing around an EET equal to $0.5 \cdot 10^{-2}$), while for Piccolo, it stops decreasing when more than 8 key bits are unknown (stabilizing around an EET equal to $1 \cdot 10^{-3}$). This decreasing behavior is mainly due to the construction phase performed by the SAT solver prior to the resolution: as the number of unknown key bits increases, this construction step becomes negligible compared to the resolution time. Since the LED system is bigger than the one of Piccolo, it is natural that the stabilization happens later for LED. Second, the values of EET obtained when enough key bits are unknown are significantly larger than the actual encryption time, even on a standard PC. This suggests that both ciphers have satisfying security margins against this type of attack. Nevertheless, the EET is larger for LED than for Piccolo (by an approximate factor 5).

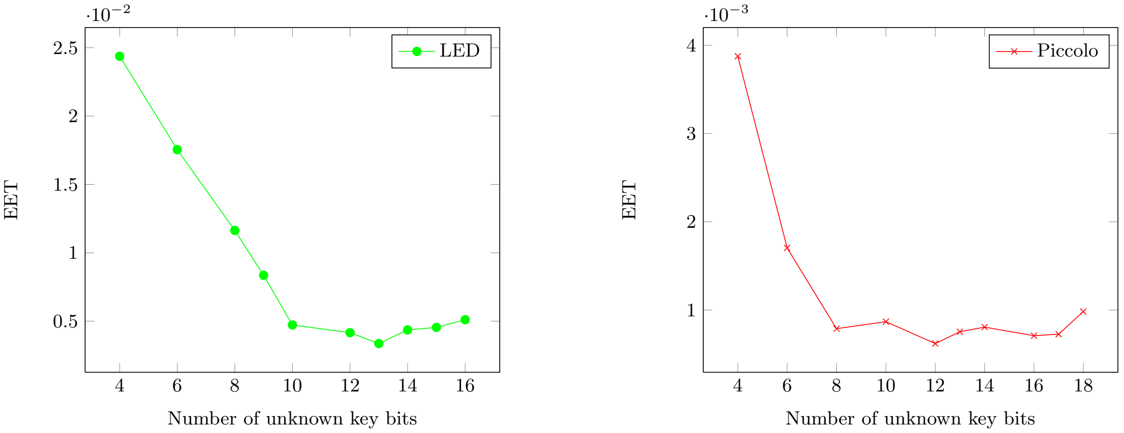

Fig. 2. EET vs number of unknown key bits for LED and Piccolo.

4.3 Attacking reduced-round versions.

As a complement to the previous experiments, we investigate the security of reduced-round versions of our target ciphers, for different number of unknown key bits. We performed experiments for a number of unknown key bits ranging from 12 to 16 for LED, and from 16 to 20 for Piccolo. This choice has been made in order to obtain equivalent ranges of resolution times for both reduced-round ciphers. Figure 3 depicts the evolution of the solving time depending on the number of rounds we have fixed. We observe that for both ciphers, the curves obtained have a similar shape, namely a highly increasing part first, and a stabilized/slowly increasing second part. This means that increasing the number of rounds beyond some limit does not provide significantly more security anymore regarding the solving time of the system. One possible interpretation of this fact is the following. The solving time of an algebraic attack primarily depends on the size of the system and the "complexity" of the equations it aims at solving (partially captured by the algebraic degree). As equations are getting more complex with the number of rounds, they reach their maximum "complexity" (and algebraic degree) at some point. From this point on, only the size of the system goes on increasing, hence explaining a slower increase of the solving time. *The full version of this work further discusses the links between the algebraic degree for the reduced-round versions of the ciphers and the bends observed in Figure 3. It also illustrates the impact of the heuristics used in SAT solvers in our experimental approach for the LED cipher.*

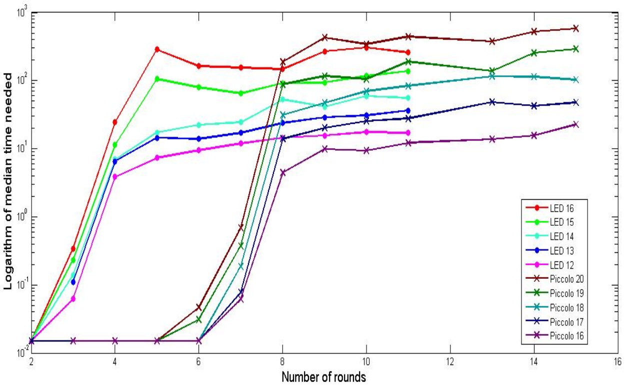

Fig. 3. Solving time for different numbers of unknown key bits and rounds (y-axis scale logarithmic).

References

1. Jean-Philippe Aumasson, Itai Dinur, Willi Meier, and Adi Shamir. Cube Testers and Key Recovery Attacks on Reduced-Round MD6 and Trivium. In Orr Dunkelman, editor, *FSE*, volume 5665 of *Lecture Notes in Computer Science*, pages 1–22. Springer, 2009.
2. Eli Biham and Adi Shamir. Differential Cryptanalysis of DES-like Cryptosystems. In Alfred Menezes and Scott A. Vanstone, editors, *CRYPTO*, volume 537 of *LNCS*, pages 2–21. Springer, 1990.
3. Alex Biryukov and Christophe De Cannière. Block Ciphers and Systems of Quadratic Equations. In Thomas Johansson, editor, *FSE*, volume 2887 of *LNCS*, pages 274–289. Springer, 2003.
4. Alex Biryukov and David Wagner. Slide Attacks. In Lars R. Knudsen, editor, *FSE*, volume 1636 of *Lecture Notes in Computer Science*, pages 245–259. Springer, 1999.
5. Christina Boura and Anne Canteaut. On the Algebraic Degree of Iterated Permutations. Finite Fields and Applications - Fq10, Gent, Belgium, 2011.
6. Christina Boura, Anne Canteaut, and Christophe De Cannière. Higher-Order Differential Properties of Keccak and *Luffa*. In Antoine Joux, editor, *FSE*, volume 6733 of *Lecture Notes in Computer Science*, pages 252–269. Springer, 2011.
7. Johannes Buchmann, Andrei Pyshkin, and Ralf-Philipp Weinmann. Block Ciphers Sensitive to Gröbner Basis Attacks. In David Pointcheval, editor, *CT-RSA*, volume 3860 of *Lecture Notes in Computer Science*, pages 313–331. Springer, 2006.
8. Carlos Cid and Gaëtan Leurent. An Analysis of the XSL Algorithm. In Bimal K. Roy, editor, *ASIACRYPT*, volume 3788 of *Lecture Notes in Computer Science*, pages 333–352. Springer, 2005.
9. Nicolas Courtois. The Dark Side of Security by Obscurity - and Cloning MiFare Classic Rail and Building Passes, Anywhere, Anytime. In Eduardo Fernández-Medina, Manu Malek, and Javier Hernando, editors, *SECRYPT*, pages 331–338. INSTICC Press, 2009.
10. Nicolas Courtois, Gregory V. Bard, and David Wagner. Algebraic and Slide Attacks on KeeLoq. In Kaisa Nyberg, editor, *FSE*, volume 5086 of *Lecture Notes in Computer Science*, pages 97–115. Springer, 2008.
11. Nicolas Courtois and Josef Pieprzyk. Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. In Yuliang Zheng, editor, *ASIACRYPT*, volume 2501 of *Lecture Notes in Computer Science*, pages 267–287. Springer, 2002.
12. Joan Daemen and Vincent Rijmen. The Wide Trail Design Strategy. In Bahram Honary, editor, *IMA Int. Conf.*, volume 2260 of *Lecture Notes in Computer Science*, pages 222–238. Springer, 2001.
13. Niklas Eén and Niklas Sörensson. An Extensible SAT-solver. In Enrico Giunchiglia and Armando Tacchella, editors, *SAT*, volume 2919 of *Lecture Notes in Computer Science*, pages 502–518. Springer, 2003.
14. Thomas Eisenbarth, Zheng Gong, Tim Guneysu, Stefan Heyse, Sebastiaan Indesteege, Stephanie Kerckhof, Francois Koeune, Tomislav Nad, Thomas Plos, Francesco Regazzoni, Francois-Xavier Standaert, and Loic van Oldeneel tot Oldenzeel. Compact Implementation and Performance Evaluation of Block Ciphers in ATtiny Devices. ECRYPT Workshop on Lightweight Cryptography, pages 71-86, Louvain-la-Neuve, Belgium, 2011.
15. Jeremy Erickson, Jintai Ding, and Chris Christensen. Algebraic Cryptanalysis of SMS4: Gröbner Basis Attack and SAT Attack Compared. In Donghoon Lee and Seokhie Hong, editors, *ICISC*, volume 5984 of *Lecture Notes in Computer Science*, pages 73–86. Springer, 2009.
16. Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED Block Cipher. In Preneel and Takagi [20], pages 326–341.
17. Lars R. Knudsen and David Wagner. Integral Cryptanalysis. In Joan Daemen and Vincent Rijmen, editors, *FSE*, volume 2365 of *Lecture Notes in Computer Science*, pages 112–127. Springer, 2002.
18. Mitsuru Matsui. Linear Cryptoanalysis Method for DES Cipher. In Tor Helleseth, editor, *EUROCRYPT*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1993.
19. Sean Murphy and Matthew J. B. Robshaw. Essential Algebraic Structure within the AES. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2002.
20. Bart Preneel and Tsuyoshi Takagi, editors. *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, volume 6917 of *Lecture Notes in Computer Science*. Springer, 2011.
21. Mathieu Renauld and François-Xavier Standaert. Algebraic Side-Channel Attacks. In Feng Bao, Moti Yung, Dongdai Lin, and Jiwu Jing, editors, *Inscrypt*, volume 6151 of *Lecture Notes in Computer Science*, pages 393–410. Springer, 2009.
22. Mathieu Renauld, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. Algebraic Side-Channel Attacks on the AES: Why Time also Matters in DPA. In Christophe Clavier and Kris Gaj, editors, *CHES*, volume 5747 of *Lecture Notes in Computer Science*, pages 97–111. Springer, 2009.
23. Claude E. Shannon. Communication Theory of Secrecy Systems. Bell System Technical Journal, vol. 28(4), page 656-715, 1949.
24. Kyoji Shibutani, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai. Piccolo: An Ultra-Lightweight Blockcipher. In Preneel and Takagi [20], pages 342–357.